

**AMENDMENTS TO THE CLAIMS**

Claim 1. (currently amended) A decoder for decoding digital information by determining a log likelihood logarithmically expressing a probability of passing a given state on the basis of a received value regarded as soft-input and decoding an input using the log likelihood, wherein the input is digital information encoded as convolutional codes, said decoder comprising:

a linear approximation means for calculating a correction term to be added to the log likelihood, the correction term being expressed in a one-dimensional function relative to a variable; wherein said variable is an absolute value of the input data; and

said linear approximation means being adapted to compute said correction term using a coefficient representing a gradient of said function for multiplying said variable, the coefficient being expressed as a power exponent of 2;

wherein said linear approximation means discards bits from the lowest bit to the k-th lowest bit of the input data when the power exponent expressing the coefficient representing the gradient of said function is expressed by  $-2-k$ .

Claims 2-3. (canceled)

Claim 4. (original) The decoder according to claim 1, wherein said linear approximation means computes said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as a power exponent of 2.

Claim 5. (original) The decoder according to claim 4, wherein said linear

approximation means computes said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as  $2^m - 1$ .

Claim 6. (original) The decoder according to claim 5, wherein said linear approximation means discards the bits from the lowest bit to the k-th lowest bit of the n-bit input data and inverts the m bits from the k-th lowest bit to the m+k-th lowest bit when the coefficient representing the gradient of said function is expressed by means of  $-2^{-k}$ .

Claim 7. (original) The decoder according to claim 1, wherein said correction term shows a positive value.

Claim 8. (original) The decoder according to claim 7, wherein said linear approximation means makes the correction term equal to 0 when a negative value is produced by computing said correction term.

Claim 9. (original) The decoder according to claim 1, wherein said log likelihood is a log expression of said probability, using the natural logarithm.

Claim 10. (previously presented) The decoder according to claim 1, further comprising:

a first probability computing means for computing for each received value a first log likelihood logarithmically expressing a first probability determined by a code output pattern and said received value;

a second probability computing means for computing for each received value a second log likelihood logarithmically expressing a second probability of getting to each state from a coding starting state in the time series;

a third probability computing means for computing for each received value a third log likelihood logarithmically expressing a third probability of getting to each state from a coding terminating state in the inverted time series; and

said second probability computing means and said third probability computing means having the linear approximation means.

Claim 11. (previously presented) The decoder according to claim 10, further comprising:

a soft-output calculation means for calculating a log soft-output logarithmically expressing the soft-output in each time slot, using said first log likelihood, said second log likelihood and said third log likelihood.

Claim 12. (original) The decoder according to claim 11, wherein said log soft-output is a logarithmic expression of said soft-output, using the natural logarithm.

Claim 13. (previously presented) The decoder according to claim 1, wherein said log likelihood is determined by computations replacing multiplications for computing the probability by logarithmic additions and additions for computing the probability by logarithmic maximum value computations and computations of said function.

Claim 14. (original) The decoder according to claim 13, wherein a maximum a posteriori probability decoding operation is conducted on the basis of the Log-BCJR algorithm.

Claim 15. (canceled)

Claim 16. (currently amended) A decoding method for decoding digital information by determining a log likelihood logarithmically expressing a probability of passing a given state on the basis of a received value regarded as soft-input and decoding an input using the log likelihood, wherein the input is digital information encoded as convolutional codes, said decoding method comprising:

a linear approximation step for calculating a correction term to be added to the log likelihood, the correction term being expressed in a one-dimensional function relative to a variable; wherein said variable is an absolute value of the input data; and

said linear approximation step being adapted to compute said correction term using a coefficient representing a gradient of said function for multiplying said variable, the coefficient being expressed as a power exponent of 2;

wherein said linear approximation step discards bits from the lowest bit to the k-th lowest bit of the input data when the power exponent expressing the coefficient representing the gradient of said function is expressed by  $-2^{-k}$ .

Claims 17-18. (canceled)

Claim 19. (original) The decoding method according to claim 16, wherein said

linear approximation step is adapted to compute said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as a power exponent of 2.

Claim 20. (original) The decoding method according to claim 19, wherein said linear approximation step is adapted to compute said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as  $2^m - 1$ .

Claim 21. (original) The decoding method according to claim 20, wherein said linear approximation step is adapted to discard the bits from the lowest bit to the k-th lowest bit of the n-bit input data and inverts the m bits from the k-th lowest bit to the m+k-th lowest bit when the coefficient representing the gradient of said function is expressed by means of  $-2^{-k}$ .

Claim 22. (original) The decoding method according to claim 16, wherein said correction term shows a positive value.

Claim 23. (original) The decoding method according to claim 22, wherein said linear approximation step is adapted to make the correction term equal to 0 when a negative value is produced by computing said correction term.

Claim 24. (original) The decoding method according to claim 16, wherein said log likelihood is a log expression of said probability, using the natural logarithm.

Claim 25. (previously presented) The decoding method according to claim 16, further comprising:

a first probability computing step for computing for each received value a first log likelihood logarithmically expressing a first probability determined by a code output pattern and said received value;

a second probability computing step for computing for each received value a second log likelihood logarithmically expressing a second probability of getting to each state from a coding starting state in the time series;

a third probability computing step for computing for each received value a third log likelihood logarithmically expressing a third probability of getting to each state from a coding terminating state in the inverted time series; and

said second probability computing step and said third probability computing step including the linear approximation steps.

Claim 26. (original) The decoding method according to claim 25, further comprising:

a soft-output calculating step for calculating a log soft-output logarithmically expressing the soft-output in each time slot, using said first log likelihood, said second log likelihood and said third log likelihood.

Claim 27. (original) The decoding method according to claim 26, wherein said log soft-output is a logarithmic expression of said soft-output, using the natural logarithm.

Claim 28. (previously presented) The decoding method according to claim 16, wherein said log likelihood is determined by computations replacing multiplications for computing the probability by logarithmic additions and additions for computing the probability by logarithmic maximum value computations and computations of said function.

Claim 29. (original) The decoding method according to claim 28, wherein a maximum a posteriori probability decoding operation is conducted on the basis of the Log-BCJR algorithm.

Claim 30. (canceled)